(2)

Can Stochastic Modelling and Analysis of Multistage Interconnection Networks Lead to Efficient Usage of Redundancy for Fault-Tolerant Design?

Amiya Bhattacharya
University of California, San Diego
La Jolla, CA 92093.

Introduction

The rapid growth in device density achieved by VLSI technology over the past decade had the attention of researchers centered around the design of array processors. The *systolic arrays* had been designed for a wide variety of applications, and consequently formal strategies for mapping algorithms onto processor arrays were developed. Use of spare or temporarily idle processing elements to achieve *fault tolerance* became an attractive aspect of VLSI array processing. The goal was to accomplish the designated task in case of transient or permanent failure of one or more processors, and to achieve a *graceful degradation* as far as possible. However, formal mapping techniques have not been extended to capture the issues of redundancy mapping. Moreover, processor interconnection topology like *shuffle-exchange* or *butterfly networks, which require global communication links, were not considered to be practical* for large on-chip design. The overhead of routing the global links to incorporate even an effective amount of redundancy is prohibitively high in terms of chip-area, signal proapgation delay and power dissipation. With the advent of the optical technology, the proposed *free-space optical interconnections* have now offered the feasibilty of having those global links established with much less overhead, making it desirable to investigate the fault-tolerant capabilities of these networks.

Two problems in this area worth further elaboration. First, the dependence structure of the algorithms implemented by processor arrays are shift-invariant and susceptible to linear tranformations, whereas that of the multistage interconnection networks and related algorithms are more complex. The scenario is even more complicated in presesnce of redundancy, usually introduced to take care of failures. A formal redundancy mapping methodology is thus sought. The second problem arise at this point, where the objective of the mapping has to be defined in terms of a trade-off between performance and fault-tolerance. For array processing, because of the uniformity of dependence structure, bandwidth (sometimes normalised with respect to the available I/O capacity) serves as a good measure. On the other hand, the amount of recomputation or available processors for system reconfiguration are reasonably good metrics for fault-tolerance. However, for the multistage networks, simple metrics like the number of paths between two communicating nodes or just the number of switches or links do not suffice. One should better use performance and reliabilty modelling to choose between design alternatives, and if possible, integrate this with the redundancy mapping technique. One way to achieve this is to pick as design metrics some measures which have some physical

UNIVERSITY OF CALIFORNIA, SAN DIEGO

BERKELEY • DAVIS • IRVINE • LOS ANGELES • RIVERSIDE • SAN DIEGO • SAN FRANCISCO

SANTA BARBARA • SANTA CRUZ

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

9500 GILMAN DRIVE
LA JOLLA, CALIFORNIA, 92093-0407

April 15, 1992

Re: Semi-annual report on ONR grant no: N00014-91-J-1017
Title: Fault Tolerance in Opto-electronic Computing
Principle Investigator: Professor Ting-Ting Y. Lin
Addressees:
Scientific Officer, Dr. Clifford G. Lau
Administrative Grants Officer
Director, Naval Research Laboratory
Defense Technical Information Center

Dear Sirs,

This letter report for the period of 1 January 1992 through 15 April 1992 constitutes the following sections: research assistants' progress, and equipment expenses.

1. Research Assistants' Progress

As stated in my previous report, two graduate students have been supported under this award since May 1991. During the past three months, Amiya Bhattacharya, a graduate student in the Computer Science and Engineering department, has been concentrating on two research issues. The first issue calls for a formal redundancy mapping methodology for fault-tolerant optical multistage interconnection architectures, whereas the second issue relates to defining a corresponding metric for evaluation that combines performance and reliability. A survey conducted on performability reveals several problems which will be discussed in the attached report.

John Comito, a graduate student in the Electrical and Computer Engineering department, focused on the fault modeling project for opto-electronic systems. He has done an extensive survey on the different component groups. The survey paper is to be submitted to Applied Optics for publication. A draft of the paper is attached.

2. Equipment expenses

Several purchases were made to facilitate the development of project. These include a NeXT 68040 upgrade board, and a NeXT Turbo Station, both are used to support student research in simulation and evaluation. The decision on another NeXT workstation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

BERKELEY • DAVIS • IRVINE • LOS ANGELES • RIVERSIDE • SAN DIEGO • SAN FRANCISCO

SANTA BARBARA • SANTA CRUZ

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

9500 GILMAN DRIVE
LA JOLLA, CALIFORNIA, 92093-0407

was made based on its integrability to the present NeXT network existing in the Computer Engineering Group.

The above two sections detailed both technical as well as budgetary issues. Immediately following this letter report, two research reports are attached. If there are issues not described or not clear, please do not hesitate to call me. Thank you.

Best regards,

Ting-Ting Lin
(619) 534-4738

significance. For example, one can use *performability,* the probability that the system performs above some performance level specified as a parameter, as one kind of such measure which combines the concepts of performance and reliabilty. Most of these measures used in the literature of fault-tolerance, being defined probabilistically or in terms of expected values, calls for using probability theory for analysis. Stochastic modelling seems to be a reasonable choice, as it is capable of capturing the distribution of message traffic and faults along with the way they interact.

Two Aspects of Design Framework

Representation of Redundancy

Traditionally, redundancy is classified into three types: *hardware redundancy, time redundancy* and *information redundancy.* However, the first two kinds can be viewed as mapping of the more basic information redundancy onto space or time. In particular, whenver a function is computed more than once, either at different site or time, the node representing that computation in the *data dependence graph* (DG) can be replicated, thus producing a graph called *redundant data-dependence graph* (RDG). For example, extra bits sent for error detection or correction may be represented by additional edges between two nodes of a bit-level RDG. The computation represented by the nodes and the data flow through the edges thus can be chosen appropriately. Effectively, information redundancy remains the abstraction which expresses itself in the physical form of either hardware or time redundancy, or as a combination of both.

Having defined the RDG in this manner, it remains to examine the directional classification of RDG. For systolic algorithms, DGs are shift-invariant, i.e. the dependence arcs do not change with respect to node positions. Canonical mapping could be applied to arrive at an SFG *signal-flow graph* from DG. But, that doesn't ensure that RDG will also be shift-invariant. They may be directional shift-invariant (DSI) or *pseudo directional shift-invariant* (PDSI). It is known that if the graph that is at least pseudo-DSI, will map to a *structurally time-invariant* (STI) graph. The goal is to investigate the restrictions that apply in different fault-tolerant arrays and networks, so that the features of RDG can be analysed better for mapping.

Performance-Reliability Modelling and Choice of Measure

Although a number of articles have appeared in the literature on performance analysis of multistage interconnection networks, very little of that can be attributed to the redundant-path fault-tolerant networks. The question is whether the results for non-redundant ones can easily be extended or not. The answer appears to be quite negative. Let us justify on this point.

First let us consider the suggestion that a simple intuitive combinatorial measure exists. The first candidate is *bandwidth* - the total number of requests that can be routed in a cycle. Of course, we consider the average value, and use expected value in theoretical derivation. But, this only gives a global measure; i.e. it doesn't show the worst possible service that can be received by a routing request, which can be a bottleneck in a parallel computation environment. Another measure can be the *largest realizable system,* i.e. the maximum number of input and outputs can be connected at a given state of faulty system.

Statement A per telecon Dr. Clifford Lau
ONR/Code 1114
Arlington, VA 22217-5000

NWW 5/22/92

Even that doesn't suffice, as no single fault disconnects a fault-tolerant network - even certain combinations of multiple faults can be tolerated. The best choice is indeed to do probabilistic analysis first, and use the expected values of performability or availability as the combinatorial metric to guide a design.

Another important issue is to understand the working model of the network. It may be *circuit-switched* or *packet-switched,* the later case has its variations of having variable size of buffer starting from none. Even when we consider the circuit-switching, the operational mode can be *synchronous* or *asynchronous.* In the synchronous case, data transfers occur in sweeps - so the set-up time and data-transfer time constitute the cycle time. Messages have to be of comparable length for this purpose. In asynchronous operation, data lengths can have a distribution, which affects the duration of an established path being held, which in turn affects the steady-state of the system. The single switch set-up time can be used as an elementary cycle time. The definition of cycle time, along with the request generation and service rates have to be normalised to do any effective comparison between different operational modes.

In any attempt for modelling, the assumptions play the most significant role. On one side, the absolute reality cannot be modelled in most cases because of the complexity involved. On the other hand, some of the simplifying assumption may cause drastic difference from the reality. One such assumption is that blocked requests are ignored and never resubmitted, thus forcing independence between requests generated in two different cycles. Early analytical and simulation work backs this assumption pointing out that the deviation is negligible at the cost of reduction of a great amount of complexity. This actually depends on the relative length of set-up and transfer time, and the effect has been analysed in a recent work.

The fact that comes in the way between simple extension of previous work to the redundant networks is that all of the previous work consistently assumed that requests are generated with uniform distribution over all possible destinations. This may be far from reality, although due to regularity, a large class of application program are expected to show some kind of stedy distribution. As a result, different section of the network may experience a significantly different load. Even if the entering traffic is symmetric, the symmetry will be lost when faults start to occur. The combined effect of these two can push the system far from the model if no modification is done to the uniform traffic assumption. A suitable modelling using renewal process can be of use in this case. In a renewal model, a number of independent and identically distributed random variables are considered having some unknown distribution. In the limiting case, some results can still be derived relating the expected values of the random varibles and their powers. These can be integrated into a combitorial metric for topology design.

Markov chain has been widely used to model the staes of the system and transitions between them. An approach to view the environment from the position of a single message has been employed - but this is unlikeky to produce the correct result in our case, because as the system starts deviating from uniformity, the transision probabilities differ from message to message. The situation requires the highest attention at this point, so that more practical modelling can be done for the fault-tolerant network. The performance figures will then be translated to design metrics.

It may be worthwhile to cite one example of what a combinatorial performance-reliabilty metric could be. In the context of a systolic array, if one decides to do each

computation more than once, possibly at different processor and at different time, and compare the results either for *concurrent error detection* or for *majority voting,* a measure called *Huang-Abraham Ratio* was proposed as $R = PBT^2/CI$, where

P = No. of PEs
B = Input Bandwidth
T = Latency
C = Gross volume of computation
I = Input volume

The Ratio can thus be represented as $R = (BT/I)(PT/C)$, where the first factor is the reciprocal of the bandwidth utilization factor (i.e. throughput measure), and the second one is the reciprocal of computational capacity usage factor (i.e. redundancy measure). The choice of this kind of metric is very much dependent on how the RDG is formulated and what kind of fault-tolerance the designer implements (e.g. in this case, RESO technique for time redundancy). The final goal of this research is to define generic performance-redundancy metrics in terms of RDG, which will offer such simplicity as Huang-Abraham Ratio, with a possible physical and intuitive relationship with performability or similar measures.

Operational Models for Analysis

The literature attempting theoretical analysis of performance of multistage interconnection networks are not based on the same ground. The variations come from the underlying modes of operation and traffic pattern. The switching strategy and buffering between stages, system timing and handling of blocked requests constitute the modes. Let us have a closer look at how they influence the performance measures.

Circuit-switching vs. Packet-switching

In a circuit-switched network, a physical path is established between the source and the destination, which is held until the message transfer is complete. In the class of delta networks digit-control routing is used, and one digit of the destinationn address is used at each stage of switch to grow the path from the source to the destination. In the case of packet-switching, a number of packets proceed through the switch stages in the same manner - the destination address being only a part of the packet which also contains the data. Patel, Thanawastien and Nelson, Kumar and Reibman, Wu and Lee considered circuit switching in their analysis, whereas the works of Dias and Jump, Kruskal, and Kumar and Jump are based on the packet switching model.

Let the network have n stages. Let the delay at each switch be t_d. If t_{select} is the switch set-up time and t_{pass} is the time taken for each packet transfer, then $t_d = t_{select} + t_{pass}$, in case of packet switching. In case of circuit switching, $t_d = t_{select}$.

The operation of unbuffered packet-switched network thus has a close resemblance to the circuit-switched operation, except for the fact that a network cycle would not be complete in the later case unless one includes the path-hold time or memory-access time

t_m. However, the model changes quite a bit with inclusion of buffer. The case has been studied by Dias and Jump. Once a packet is buffered at a switch, its transfer time to the next stage involves t_{select} plus a multiple of t_{pass} depending on when it gets its turn to be propagated. In case the buffer is full, the backward propagation delay in informing the predecessor switch can also be included in t_{pass}.

System Timing

The operation of the network can be synchronous or asynchronous. In general it is easier to visualise and model the synchronous operation. But this would not be a practical choice unless there is uniformity over the processor and memory operation times. For example, in their analysis of unbuffered packet-switched nework, Kumar and Jump considered all switches to be synchronised by a single clock, and that only one packet could be transfered between stages by a link of capacity one in one cycle. Obviously, t_d is an ideal choice for the clock cycle time. In case the switches are clocked in the circuit-switched model, one can define the clock cycle similarly. This definition has been used in the analysis by Wu and Lee, and should not be confused with the network cycle time used by Patel, Thanawstien and Nelson, and Kumar and Reibman. In synchronous mode of circuit switching, this is the time for one sweep of transfer from the sources to the destinations. All the sources submit their requests at the beginning of this cycle. Thus, network cycle time T_c can be written as $T_c = t_p + nt_d + t_m$, where t_p is the time taken by the processor to generate a request, and t_m is the memory access time. In asynchronous mode, T_c can be replaced by the data-transfer time for each individual request, the length of which are different.

Blocked Request Handling

Once more than one requests collide for an output port of a switch, one of them is randomly chosen for transfer, a number of them are queued if buffer is available, and the rest are discarded. Different assumptions have been adopted regarding the fate of the discarded requests. Patel considered the option of not doing a resubmission, and argued for the fact that the result differs only by a small amount in this analysis. Resubmission and its effect have been considered later by Kruskal, Thanawastien and Nelson, and Wu and Lee. The later also considered the effect of the intermediates - drop and hold strategies and their combinations. Kumar and Reibman neglected the setup time and thus avoided the issue.

Reference

T. Lang.
Interconnections between Processors and Memory modules using the Shuffle-Exchange Networks.
IEEE Transaction on Computers, August 1980.

P. Y. Chen, D. H. Lawrie , P. C. Yew and D. A. Padua.

Interconnection Networks using Shuffles.
IEEE Computer, December 1981.

D. M.. Dias and J. R. Jump.
Analysis and Simulation of Buffered Delta Networks .
IEEE Transactions on Computers, April 1981 .

S. Thanawastien and V. P. Nelson.
Interference Analysis of Shuffle-Exchange Networks .
IEEE transactions on Computers, August 1981.

J.H. Patel .
Performance of Processor-Memory Interconnections for Multiprocessors .
IEEE Transactions on Computers, October 1981.

D. M. Dias and J. R. Jump .
Packet Switching Interconnection Networks for Modular Systems.
IEEE Computer, December 1981.

C. P. Kruskal and M. Snir .
The Performance of Multistage Interconnection Networks for Multiprocessors.
IEEE Transactions on Computers, December 1983.

K. Padmanabhan and D. H. Lawrie.
A Class of Redundant Path Multistage Interconnection Networks .
IEEE Transactions on Computers, December 1983.

J. W. Goodman, F. J. Leonberger, S. Y. Kung and R. A. Athale .
Optical Interconnections for VLSI Systems.
Proceedings of the IEEE, July 1984.

J. A. B. Fortes and C. S. Raghavendra.
Gracefully Degradable Processor Arrays.
IEEE Transactions on Computers, November 1985.

M. Kumar and J. R. Jump.
Performance of Unbuffered Shuffle-Exchange Networks.
IEEE Transactions on Computers, June 1986.

W. H. Wu, L. A. Bergman, A. R. Johnston, C. C. Guest, S.C. Esener, P. K. L. Yu, M. R. Feldman and S. H. Lee .
Implementation of Optical Interconnections for VLSI .
IEEE Transactions on Electronic Devices , March 1987.

A. A. Sawchuk, B. K. Jenkins, C. S. Raghavendra and A. Varma.

Optical Crossbar Networks.
IEEE Computer, June 1987.

M. R. Feldman and C. C. Guest.
Computer Generated Holographic Optical Elemments for Optical Interconnection of Very Large Scale Intergrated Circuits.
Applied Optics, October 15, 1987.

T. Feng.
A Survey of Interconnection Networks.
IEEE Computer, December 1987.

S. W. Chan and C. L. Wey.
The Design of Concurrent Error Diagnosable Systolic Arrays for Band Matrix Multiplications.
IEEE Transactions on Computer-Aided Design, January 1988.

R. M. Smith, K. S. trivedi and A. V. Ramesh.
Performability Analysis : Measures, an Algorithm and a Case Study.
IEEE Transactions on Computers, April 1988.

M. R. Feldman, S. C. Esener, C. C. Guest and S. H. Lee .
Comparison between Optical and Electrical Interconnects based on Power and Speed Considerations.
Applied Optics, May 1 , 1988.

T. Leighton and B. Maggs.
Expanders Might Be Practical : Fast Algorithms for Routing around Faults in Multibutterflies .
Proc. 30th Annual Symposium on Foundations of Computer Science, IEEE, October 1989.

V. P. Kumar and A. L. Reibman.
Failure Dependent Performance Analysis of a Fault-tolerant Multistage Interconnection Networks.
IEEE Transactions on Computers, December 1989.

T. Leighton, D. Lisinski and B. Maggs.
Empirical Evaluation of Randomly-Wired Multistage Networks.
Proc. International Conference on Computer Design : VLSI in Computers and Processors, IEEE, September 1990.

S. Arora , T. Leighton and B. Maggs.
On-Line Algorithms for Path-Selection in a Nonblocking Network.
Proc. 22nd Annual ACM Symposium on Theory of Computing, May 1990.

M. J. Taylor and J. E. Midwinter.
Optically Interconnected Switching Networks.
Journal of Lightwave Technology, June 1991.

C. L. Yu and M. Lee.
Performance Analysis of Multistage Interconnection Network Configurations and Operations.
IEEE Transactions on Computers, January 1992.